VOLUMEN II NO.1 JULIO 198

ESPECIAL MICRODRIVES

INFORMACION SOBRE EL CLUB

La integración en la asociación QLave se hace por suscripción anual o semestral. QLave publica mensualmente el boletín de los socios.

Más información sobre la asociación puede obtenerse desde la secretaría del Club.

Para ser miembro de QLave se requiere estar interesado por el ordenador personal Sinclair QL.

El Club mantiene una librería de software. Una lista de los programas existentes en la librería se publicará de vez en cuando para así ir actualizándola. Los programas que se quieran aportar o sacar de la librería se deben notificar al encargado de la misma.

Presidente	Secretario	Librero
Serafín Olcoz	Juan Palacio	Angel Asín.

La correspondencia debe enviarse a: QLave / Apartado de correos 403 / 50080 - Zaragoza

Contribuciones a QLave

Las contribuciones a QLave deben ser archivos de QUILL en cartuchos de microdrive preferiblemente. Los cartuchos se devuelven a vuelta de correo.

Los programas cuya extensión no sea muy grande se incluirán en la revista, para aquellos de gran extensión pasarán a formar parte de la librería.

Los programas que se envíen deben adjuntar una descripción de los mismos y de su funcionamiento.

==========

Se prohíbe la reproducción total o parcial del contenido de esta revista.

QLave no se hace responsable del contenido de los artículos o comentarios que aparezcan firmados por sus correspondientes autores.

Sinclair, QL, QDOS, ZX microdrive son marcas registradas de Sinclair Research Ltd.

Portada de Fabio Licer Navarro de Letosa.

EDITORIAL

El pasado dia 30 de Junio me acerqué nuevamente a las oficinas del registro de asociaciones del Gobierno Civil de Zaragoza para interesarme por la "salud" de nuestros estatutos. Como ya esperaba no estaban y seguian retenidos en Madrid.

El motivo de que los estatutos estén en Madrid se debe a que el C.E.I.U.Q.L. es de ámbito nacional y por tanto excede su aprobación a las competencias del Gobierno Civil de Zaragoza.

Aunque quizá no sea un gran consuelo, el motivo de la retención de los estatutos en Madrid se debe únicamente a problemas burocráticos ya que por lo visto desde principios de este año hay muchas asociaciones, entre las cuales está ULave, con sus estatutos pendientes de aprobación .

Desde estas páginas quiero dar la enhorabuena a Tony Tebby ya que su proyecto va por muy buen camino. QUMP LTD. se ha puesto de acuerdo con la empresa SANDY de MILAN para que ésta dirija la producción y el marketing del QLT. Según dice Tony, Sandy es una compañia capaz de llevar al mercado productos en muy corto espacio de tiempo y fabricará el QLT en cantidades suficientes para abarcar el mercado mundial. Por ahora no se conoce la politica de mercado de Sandy pero según parece y como ya indicaba el mès pasado el QLT, la màquina base (512 Kbyte RAM, un disco de 720 Kbyte 3.5") costará alrededor de 500 Libras incluyendo el VAT (IVA) en el Reino Unido, con precios equivalentes en el resto del mundo.

Tony espera, si no hay cambios, que el QLT esté en producción en otoño de este mismo año. Ojalá que asi sea.

Los cambios màs importantes que en las últimas semanas se han realizado en el QLT han

sido la mejora y simplificación de la capacidad de expansión del nuevo ordenador.

El QLT intenta ser una versión de alta calidad del ordenador Sinclair QL (High QuaLiTy). Como el producto aún está en desarrollo sus características pueden estar sujetas a cambios pero voy a daros una primicia:

La forma del QLT será una "caja" en la que van el procesador y el monitor juntos y el teclado aparte. El tamaño está determinado por la capacidad de conectar las tarjetas de expansión de tamaño Eurocard así como de incorporar hasta dos discos floppy de 3.5" y un disco duro también de 3.5". El monitor será de 12" o 14". La parte del procesador del QLT está dividida en tres unidades: la MPU (MC68000) y los servicios periféricos, el procesador de display y por último la RAM. La RAM se presenta en módulos de 512 Kbytes o 2 Mbytes. La expansión primaria es posible empleando uno de los varios bus_conectores reducidos de 48 pin ROMPer (ROM/Peripheral). La expansión de procesadores se puede realizar por medio de ROMPers junto con el bus de RAM.

Por otra parte, respecto al THOR de Cambridge Systems Technology (C.S.T.) una grata noticia y es que ya se pueden hacer las reservas para adquirirlo en septiembre por 550 Libras + IVA (con un sólo floppy). Su configuración es la de un teclado IBM y una caja con un sólo floppy en la configuración base (sin monitor) con 640 K de memoria RAM, 128 K EPROM, QDOS, consta de cuatro puertas para ROM, otra para impresora en paralelo y otra para ratón, por supuesto el software es compatible con el QL. Otras configuraciones posibles son: con dos floppy (650 Libras + IVA),con un floppy y un disco Winchester de 20 MB (1350 Libras + IVA).

El "QL" THOR se presentará en la Muestra del mundo de las Computadoras del próximo Septiembre (Personal Computer World Show) y Eidersoft estará en el Stand 3130.

Respecto a la PIRATERIA, EIDERSOFT pagará una recompensa a cualquiera que de información sobre piratas de software, así que ojo porque esto va a parecerse al lejano oeste.

Serafín Díaz

NOVEDADES

COMPWARE

Hardware

Sinclair QL (Psion 2 packages/ 5 free games)..... £199-99

Astracom 1000 Auto answer/dial modem (call for data sheet)..... £198-00
Microvitec CUB 14" high res. colour monitor with cable & plynth. £289-00
Phillips Monochrome 12" monitor with cable: Green screen... £109-99
(Tilt plynth for Phillips £9.99) Amber screen... £114-99

Q+4, 4 slot expansion module (CST)..... £199-99
Expansion module (Sinclair QL) power supply..... £ 25-00

512K Expandaram with through connector (for Cumana disc I/F).... £125-00

Cumana disc interface ..... £ 89-00
SuperQboard with disc and printer interface, ROM utilities and RAM:
SuperQboard £135 / SuperQboard + 256K £214 / SuperQboard + 512K £249

3.5" 720K single drive (built in psu)..... £141-00
3.5" 720K dual drives (built in psu)..... £221-00

CP80 80 colum dot matrix printer..... £225-00
Kaga Taxan Letter Qality Matrix printer..... £309-00
Daisy Step 2000 Daisywheel printer (used to print $a^2 = b^3$ this!) £279-00

3.5" floppy (DS/DD) unbranded 3M in library case.....(box of 10) £ 29-00
3.5" lockable disc storage box: For 40 discs £14, For 80 discs £ 18-00

Computer Cleaners (stop lock-ups): single mains plug (5A).....£ 14-00
trailing 4way power block(13A)£ 24-00

4 microdrives in plastic wallet (as with QL)..... £ 7-99
Microdrive storage box (holds 20 cartridges)..... £ 5-99
QL serial cable..... £15-00 Parallel Printer cable..... £ 15-00
QL Centronics interface..... £ 38-00

Business Software

Cash Trader..... £69-00 Entrepreneur..... £39-00
Project Planner..... £39-00 Decision Maker..... £39-00

Software Languages & Utilities

QL MDV Toolkit (Extensions) £ 9-99 QL Assembly Lang MDV Toolkit. £29-99
Sinclair/Tony Tebby Toolkit £24-00 Sinclair Technical Guide..... £14-00
Supercharge (Basic compiler)£59-00 Computer One Assembler..... £29-00
Computer One Pascal..... £29-00 Metacomco Assembler..... £39-00
Metacomco C..... £99-00 Adder QL Assembler..... £34-00
Boot128k..... £ 4-99 '1 to 1 Dump' (Screen dump).. £ 4-99
Sprite Designer..... £ 9-00 RAM disc on microdrive..... £14-95
Microdrive Copycat..... £ 9-99

Software Games

QL Backgammon..... £12-00 Hopper (Microdeal)..... £14-00
Hungry Harry..... £10-00 QL Caverns (Sinclair)..... £14-00

PRICES INCLUDE VAT AND DELIVERY
WRITE OR TELEPHONE FOR FURTHER DETAILS OR GENERAL HELP

UK residents: Add £2.50 per order if you require Cash On Delivery but note that your total order value must not exceed £347.50 to use the COD service.

COMPWARE
57 REPTON DRIVE
HASLINGTON
CREWE CW1 1SA
Tel: 270 582301

OFERTA ESPECIAL

Monitor de alta resolución, de 12 pulgadas, con cable para el QL . Es el monitor OFICIAL SINCLAIR para el QL y está disponoble por un tiempo limitado.

Precio superespecial (mientras haya stock)..... 199 £ incluyendo VAT y reparto (12 meses de garantía)

Precio normal ..... 299 £

ALSI comercial, S. A.

Nicolás Usera, 10 - 28026 MADRID - TELEFONO 475.43.39

HARDWARE & SOFTWARE

Tenemos el agrado de ofrecerles los siguientes productos informáticos:

Ordenador Sinclair QL.....P.V.P. 49.836
Ordenador Sinclair QL + Monitor fósforo.....P.V.P. 73.136
Ordenador Sinclair QL + Monitor fósforo + impresora .....
.....P.V.P. 129.920
Unidad de disco 1 megabyte QL.....P.V.P. 69.300
Disquetera 3 1/2 " capacidad 100 discos .....P.V.P. 5.000

Así mismo, ofrecemos la más amplia gama de periféricos para el mencionado ordenador: cables (serie y paralelo), disKettes virgenes 3 1/2 ", cables monitor, papel (facturas, recibos, pautado, pegatinas) etc.

OFERTA ESPECIAL : comprando un equipo completo del ordenador QL, regalamos dos de nuestros reconocidos programas de gestión.

También ofrecemos, con el precio a consultar, ordenador AMSTRAD PCW 8256, AMSTRAD 6128 con monitor verde o en color; ordenador ATARI 520 ST, ATARI 1040 ST.

Disponemos de todo tipo de programas de gestión y utilidades (pascal, compilador basic, "C", etc.) para AMSTRAD y ATARI.

PREGUNTAS

1 - ¿Cómo es posible obtener una alimentación para el ordenador que además permita el uso de SLOTS de EXPANSION para conectar placas en paralelo simultáneamente (ya sea las que se venden o bien las que uno mismo desee diseñar, como el interface con VIA 6522)?, porque el sistema de por sí tiene una capacidad limitada de potencia.

2 - ¿Es posible que los programas de la librería se puedan obtener en formato disco?. Esto lo pregunto porque ya sabemos que el microdrive tiene sus limitaciones y la verdad es que salvo programas que me han interesado y he comprado, el uso del sistema para trabajo lo hago exclusivamente con disquete de 3 1/2 pulgadas.

3 - Con referencia a las unidades de disco y sus interfaces, ¿existe algún problema de compatibilidad para la lectura y grabación de programas en los distintos sistemas que se venden?

Antonio Molina Morena
BARCELONA

****

Cualquier fuente de alimentación estabilizada puede ser encontrada en el mercado basta con que sepas la potencia que vas a precisar y sustituyas ésta por la que viene con el QL de fábrica.

Por supuesto que los programas de la librería pueden ser obtenidos en disco sin embargo se considera que puesto que todo el mundo tiene microdrives y su tamaño para envío por carta es muy reducido se optó por esta alternativa. Sin embargo si los deseas en disco no tienes más que remitirnos uno vacío y te salvaremos la librería.

No existen problemas de compatibilidad ya que el formato es idéntico para los discos comercializados por las diferentes casas.

Slave.

1.- En el número 4 en el programa de la página 15 de volcado de pantalla me da errores en las líneas de DATAS.

2.- Me gustaría informarme acerca del programa SUPERCHARGE.

3.- Quisiera saber si hay algún método que nos permita conocer la memoria disponible en cada momento en el ordenador ya que ando un poco confuso con lo que aparece en el número 5 página 18, a mí me da la resta: ordenador vacío 87552; con un programa que ocupa 18559. Una vez cargado el programa hago un NEW y quedan 85504 y con el programa cargado y haciendo CLEAR 59904, por lo que cada vez sale una cantidad distinta que no baja nunca de los 25K. Sin embargo este programa en una versión en Basic Microsoft y que es más larga ocupa 22K.

4.- Han aparecido en los periódicos unas ofertas de Investrónica y me gustaría saber si podeis conseguir alguna de cartuchos para el ndv_ ya que aquí en La Coruña es imposible conseguirlos y en dichos anuncios figuran ofertas de los mismos.

Manuel Angel Alonso Pica.
La Coruña.

==========

Efectivamente el checksum de la línea 500 del programa BASIC debe ser 134976.

El programa SUPERCHARGE es un compilador de SuperBasic y bastante bueno pues solo tiene algún pequeño error que con alguna pequeña alteración en el programa lo compila perfectamente, decirte que usa la zona de pantalla como memoria de compilación y que el programa en sí es bastante grande lo que hace que programas muy largos no sean compilables a ,nó ser que poseas expansión de memoria.

La resta que indicamos en el número 5 es correcta salvo en 200 bytes que deberían de ser restados además al primer peek. Decirte que desde luego que la memoria libre no tiene mucho que ver con la longitud del programa en microdrive ya que el QL sigue un sistema de apuntadores para sus programas BASIC en la máquina que no se siguen lógicamente en el microdrive, por lo que cuanto mayor sea el programa mayor será también la diferencia de tamaño entre el programa en cartucho y en memoria. Por otra parte indicarte el sistema operativo usa zonas de memoria para asignación de canales, dependiendo de los que abramos, sistemas de bloques como memoria intermedia para lectura de ficheros., por ello el primer NEW realizado no coincide con la memoria inicial pero todos los que hagas después coincidirán y darán una cantidad fija de memoria. CLEAR también da una cantidad distinta ya que limpia parte de las asignaciones de variables y canales hechas por el programa. Así la resta tal como la indicamos y menos 200 bytes da la memoria libre real.

De momento no nos hemos planteado esta posibilidad de actuar como vendedores de microdrives aunque desde luego en el momento en que lo veamos factible os lo haremos saber.

Qlave

1.- ¿ Como podria conseguirse que en el ordenador sonara al pulsar las teclas y al autorepetirse, aun funcionando otras tarèas (igual que el SPECTRUM) ?. De ser con algùn programa en multitarèa, ¿ podriais con el tiempo publicar su listado?

2.- Quisiera ampliar la memoria a 512 Kbytes. ¿ No hay peligro de sobrecalentarse el 7806 (Regulador de tensiòn.) y deteriorarse con ello todo el ORDENADOR ?. Tengo oido que algunos SPECTRUM de 16 Kbytes se deterioraron al ampliarlos a 48 Kbytes de RAM y al reparalos les aumentaban el radiador del 7806.

3.- En el ùltimo nùmero veo que la lista de productos de COMPWARE està muy disminuida. Mi pregunta es: ¿ si èsta se refiere a las novedades o es en cambio que COMPWARE disminuye su oferta ?.

4.- ¿ Què hace el JOYSTICK, sustituye algunas teclas en concreto ?. ¿ Funcionaria el AJADREZ de PSION con JOYSTICK ?. De ser positivo¡ ¿ Què Joystick se recomendaria ?.

José Manuel Nùñez Carrillo
SALAMANCA

xxxxxxxxxxxxxxxxxxxx

Nosotros no conocemos ninguna variable del sistema cuya alteraciòn permitira lo que quieres. Lògicamente si que seria posible lograr algo parecido mediante una rutina de polling que explorase continuamente el teclado y que en caso de pulsaciòn produjese un BEEP.

Hasta ahora no ha surgido ninguna pega de ninguna clase con las diferentes ampliaciones de memoria y por la experiencia se te puede decir que si estàs pensando en realizar la ampliaciòn que lo hagas ya que las ventajas son notables.

La lista de COMPWARE no ha disminuido es màs creo que ha aumentado algo. Lo que ocurre, como tu mismo planteas, es que lo que se publica son ùnicamente las novedades dando porsupuesto que lo que no se modifica en la oferta de COMPWARE, se mantiene.

Las salidas de JOYSTICK sustituye efectivamente a una fila de teclas. Para màs informaciòn ver el articulo existente en este mismo nùmero. El Ajedrez de PSION funciona con JOYSTICK. Hay innumerables marcas en èl mercado, el màs popular puede ser el Quickshot II. Si bien has de tener en cuenta que es preciso comprar un àdaptador para la peculiar salida del QL.

QLave

... sobre los joysticks

El QL tiene en su parte posterior dos conectores para joystick (CTL1 y CTL2). En principio, y según aparece en la Guia del Usuario, desde cada conector, puede conseguirse la derivación de 5 teclas. En el caso de usar CTL1, las teclas que se derivan son: F1, F2, F3, F4 y F5. Si usamos CTL2, las teclas son las correspondientes a los cuatro cursores de dirección, y el espacio. Esto es: si tenemos conectado un joystick en CTL2 obtendremos el mismo resultado pulsando el cursor hacia la derecha, que moviendo el brazo del joystick en la misma dirección; si tuviéramos el joystick en CTL1, el movimiento hacia la derecha del mismo actuaría igual que la tecla F3.

	CTL1	CTL2
JOYSTICK	Tecla a la que suple	
Hacia arriba	cursor arriba	F4
Hacia abajo	cursor abajo	F2
a la izquierda	cursor izda.	F1
a la derecha	cursor derecha	F3
disparos	espacios	F5

Así pues, el testear desde un programa la posición de un joystick es muy sencilla: basta con comprobar si F1, F2, F3, F4 o F5 están pulsados (caso de usar CTL2) ó los cursores y el espacio (si se emplea CTL1); o ambos a la vez, para el caso de que hubiera dos joysticks conectados a la vez.

Posiblemente, la forma más sencilla de detectar cuál de estas teclas se encuentra pulsada es con el comando KEYROW. Tal y como se explica en la página 33 de la Guia de referencia de palabras clave, es una función que investiga el estado que tienen en un momento dado una fila de teclas. La función KEYROW necesita un parámetro, que es la fila de teclas que se va a testear, y devuelve un valor entero comprendido dentro de la gama 0 a 255, que es una representación en binario de las teclas que se pulsaron en la fila elegida.

En la matriz del teclado del QL, las filas 0 y 1 son:

Columna	1	2	4	8	16	32	64	128
Fila								
0	F4	F1	5	F2	F3	F5	4	7
1	ENTER	izda.	arriba	ESC.	dcha.	ñ	espacio	abajo

Puede comprobarse que las teclas que se derivan en CIL1 están todas dentro de la fila 1 del teclado, y que las que se derivan en CIL2 pertenecen a la fila 7. Por eso la función KEYROW (0) nos servirá para leelear la posición de un joystick conectado en CIL2 y KEYROW (1) para indagar la de uno conectado en CIL1.

Así por ejemplo, si tenemos un joystick conectado a CIL1 del cual tengamos pulsado el interruptor del disparo, al preguntar por la posición del joystick (" PRINT KEYROW(1) ") nos contestará: "64", ya que es el valor que correspondería según aparece en el esquema de la matriz del teclado para la tecla "espacio" (equivalente al disparador del joystick). O dicho de otra manera, el CL al indagar en la fila 1, extrae la siguiente configuración: 0 0 0 0 0 1 0, que invertidanola nos da el número binario: 01000000 = 64.

Una cualidad importante de la función KEYROW es el poder detectar la pulsación de varias teclas de la misma fila.

Si en el caso anterior, en lugar de pulsar sólo el disparador, presionáramos simultáneamente el disparador y el brazo del joystick hacia arriba, al preguntarle al CL por la posición del Joystick (" PRINT KEYROW(1) ") nos devolvería el número "68", ya que tal y como aparece en la rejilla antes representada, el valor del disparo es 64, y el del cursor hacia arriba es 4; 64+4 = 68. O lo que es lo mismo, la configuración que obtiene ahora el CL de la fila 1 del teclado es: 00100010 que al invertirse nos da el número 68 en binario.

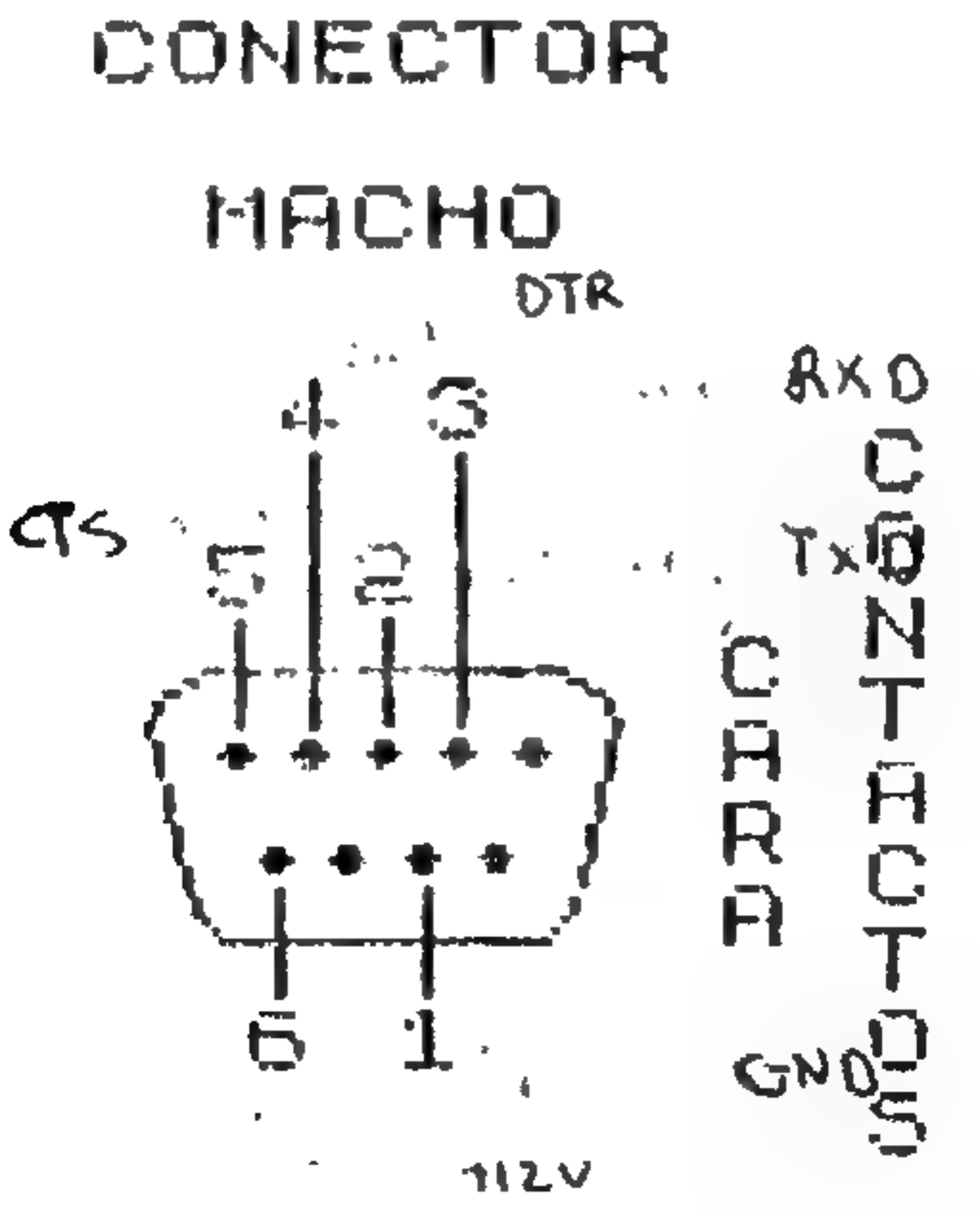
En la fig. 1 está representado el conector que se acopla a CIL1 o CIL2, para la conexión de un joystick. Dicho conector tiene 6 patillas, que en la figura se han numerado de 1 al 6. La patilla 1 es portadora de una señal eléctrica. Si dicha señal, después de pasar por el mando, retorna por una u otra de las 5 patillas restantes, indicará al QL la posición del joystik, tal y como se muestra en el esquema siguiente:

1 2 3 4 5 6

Disparo.
Hacia arriba.
Hacia abajo.
Hacia la derecha.
Hacia la izquierda.

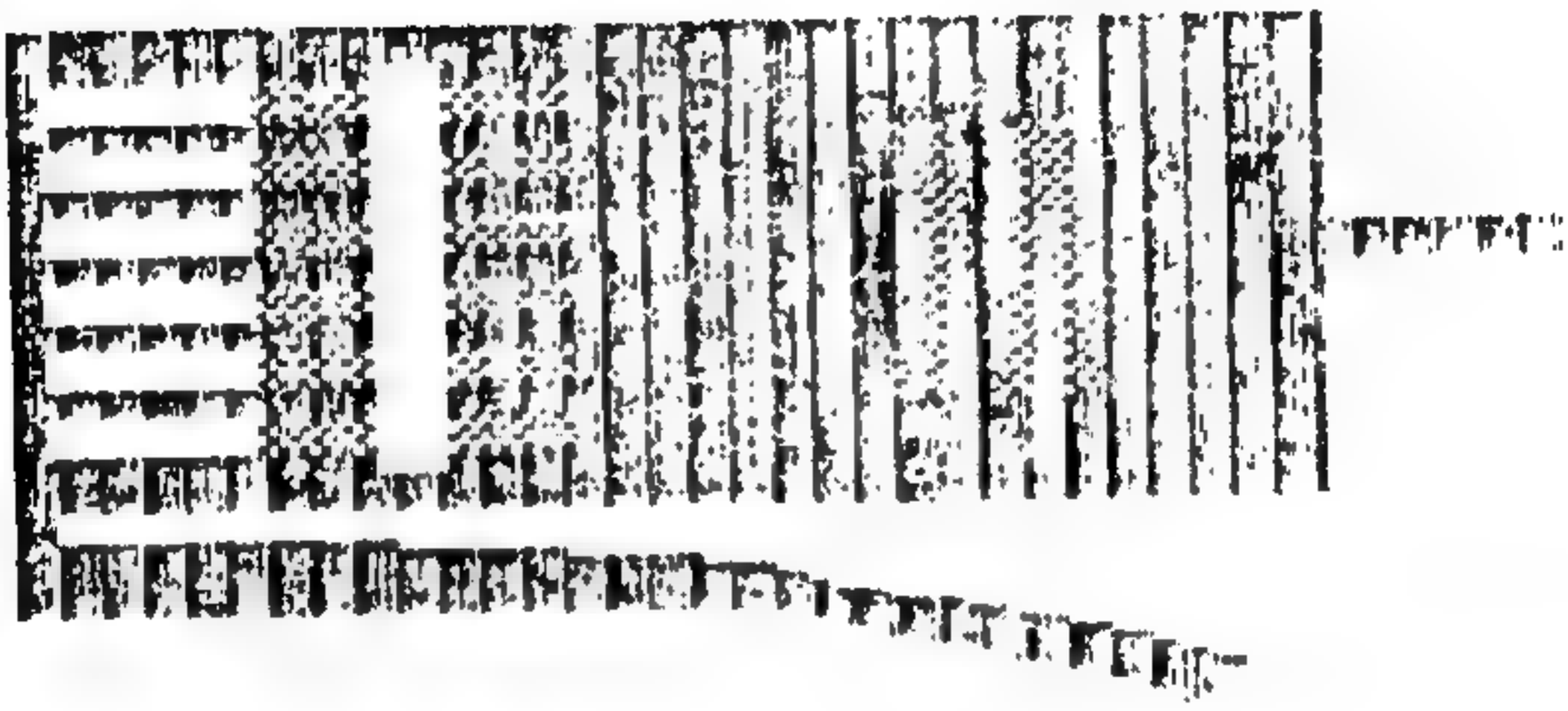
Por lo tanto, el funcionamiento interno de un joystick es muy simple: la señal que recibe por un cable (1), la retorna por uno de otros cinco posibles, según cuál de las 5 posibilidades del joystick esté activada (cuatro direcciones y disparo). En sí, un joystik no es mas que una caja con 5 pulsadores.

En la fig. 2 aparece la relación de las patillas de la clavija adoptada por Sinclair con las correspondientes, de la clavija "standard" para joysticks. (La clavija standard está vista por su parte frontal).



Juan Palacio.

W A N T E D



Para el desarrollo de un Kit para el QL necesitamos conectores como el de la fotografía, que puedan conectarse en las salidas de Joystick.
Rogamos a aquellos socios que sepan dónde adquirir dichos conertores nos lo hagan saber (Olave / Apartado de correos 403 / 50080 - Zaragoza).

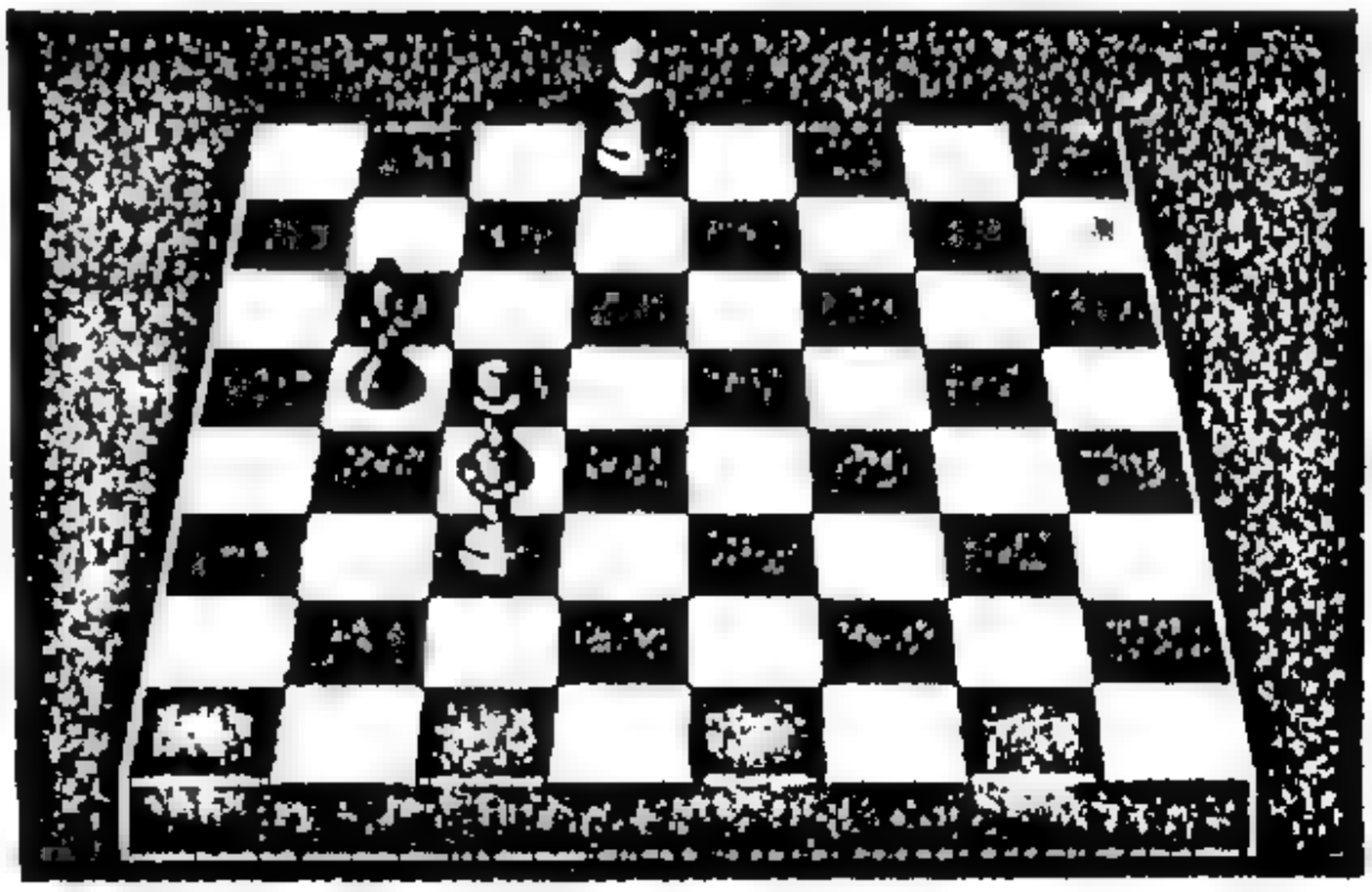
DUELO SL - - - - - AMSTRAD

Los días 3 y 4 de Julio se celebró en:
A.R.M.
C/ San Juan de la Cruz nº 19
ZARAGOZA
una partida de ajedrez entre el PCW8256 de AMSTRAD y el SL resultando
(como no) vencedor en actas confrontaciones el SL con su CHESS
de PSION ; AMSTRAD con el último de los CHESS diseñados para él.
Como muestra de estas partidas os las ofrecemos reproducidas junto con
unas pantallas del MATE.

SL BLANCAS

1	D2-D4		
1	. . .	D7-D5	
2	C2-C4		
2	. . .	E7-E6	
3	B1-C3		
3	. . .	B8-C6	
4	G1-F3		
4	. . .	F8-B4	
5	C4xD5		
5	. . .	E6xD5	
6	C1-G5		
6	. . .	F7-F6	
7	G5-F4		
7	. . .	G7-G5	
8	F4-E3		
8	. . .	C8-E6	
9	A2-A3		
9	. . .	B4xC3	
10	D2xC3		
10	. . .	D8-E7	
11	D1-B3		
11	. . .	E8-C8	
12	A1-D1		
12	. . .	C6-A5	
13	B3-B4		
13	. . .	E7xB4	
14	A3xB4		
14	. . .	A5-C4	
15	E3-C1		
15	. . .	C8-B8	
16	E2-E3		
16	. . .	E6-F5	
17	B1-A1		
17	. . .	G5-G4	
18	F3-H4		
18	. . .	F5-C2	
19	F1xC4		
19	. . .	B5xC4	
20	E1-G1		
20	. . .	H7-H5	
21	F1-E1		
21	. . .	D8-E8	
22	C1-D2		
22	. . .	A7-A6	
23	A1-A5		
23	. . .	C2-D3	
24	D4-D5		
24	. . .	A6xB5	
25	E1-A1		
25	. . .	B8-C8	
26	A5xB5		
26	. . .	B7-B6	
27	A1-A8		
27	. . .	C8-B7	
28	B5-B5		
28	. . .	B7-C6	

29	D5-C5		
29	. . .	B6xC5	
30	A8xE8		
30	. . .	C5xD4	
31	E3xD4		
31	. . .	C6-B5	
32	F2-F3		
32	. . .	B3-C2	
33	G1-F2		
33	. . .	B5-B6	
34	F2-G3		
34	. . .	G4xF3	
35	H4xF3		
35	. . .	C2-B3	
36	B2-A3		
36	. . .	B6-C6	
37	A3-E7		
37	. . .	C6-B7	
38	E7xF6		
38	. . .	B7xE8	
39	F6xH8		
39	. . .	G8-E7	
40	F3-E5		
40	. . .	E7-B5	
41	E5xB3		
41	. . .	E4xB3	
42	G3-F3		
42	. . .	B5xC3	
43	F3-E3		
43	. . .	C3-B5	
44	E3-D2		
44	. . .	E8-B7	
45	G2-G3		
45	. . .	B5-B4	
46	H8-G7		
46	. . .	B7-C6	
47	G7-F8		
47	. . .	C6-B5	
48	H2-H3		
48	. . .	B5-C4	
49	F8xB4		
49	. . .	C4xB4	
50	D2xB3		
50	. . .	C7-C6	
51	G3-G4		
51	. . .	H5xG4	
52	H3xG4		
52	. . .	C6-C5	
53	G4-G5		
53	. . .	C5-C4	
54	B3-C2		
54	. . .	B4-B5	
55	G5-G6		
55	. . .	B5-B4	
56	G6-G7		
56	. . .	B4-B5	
57	G7-G8/Q		
57	. . .	B5-C6	
58	G8-E6		
58	. . .	C6-C7	
59	B4-B5		
59	. . .	C7-B8	
60	B5-B6		
60	. . .	C4-C3	
61	E6-F7		
61	. . .	B8-A8	
62	B6-B7		
62	. . .	A8-B7	
63	B7-B8/Q		
63	. . .	B7-A6	
64	C2xC3		
64	. . .	A6-B5	
65	F7-C4		

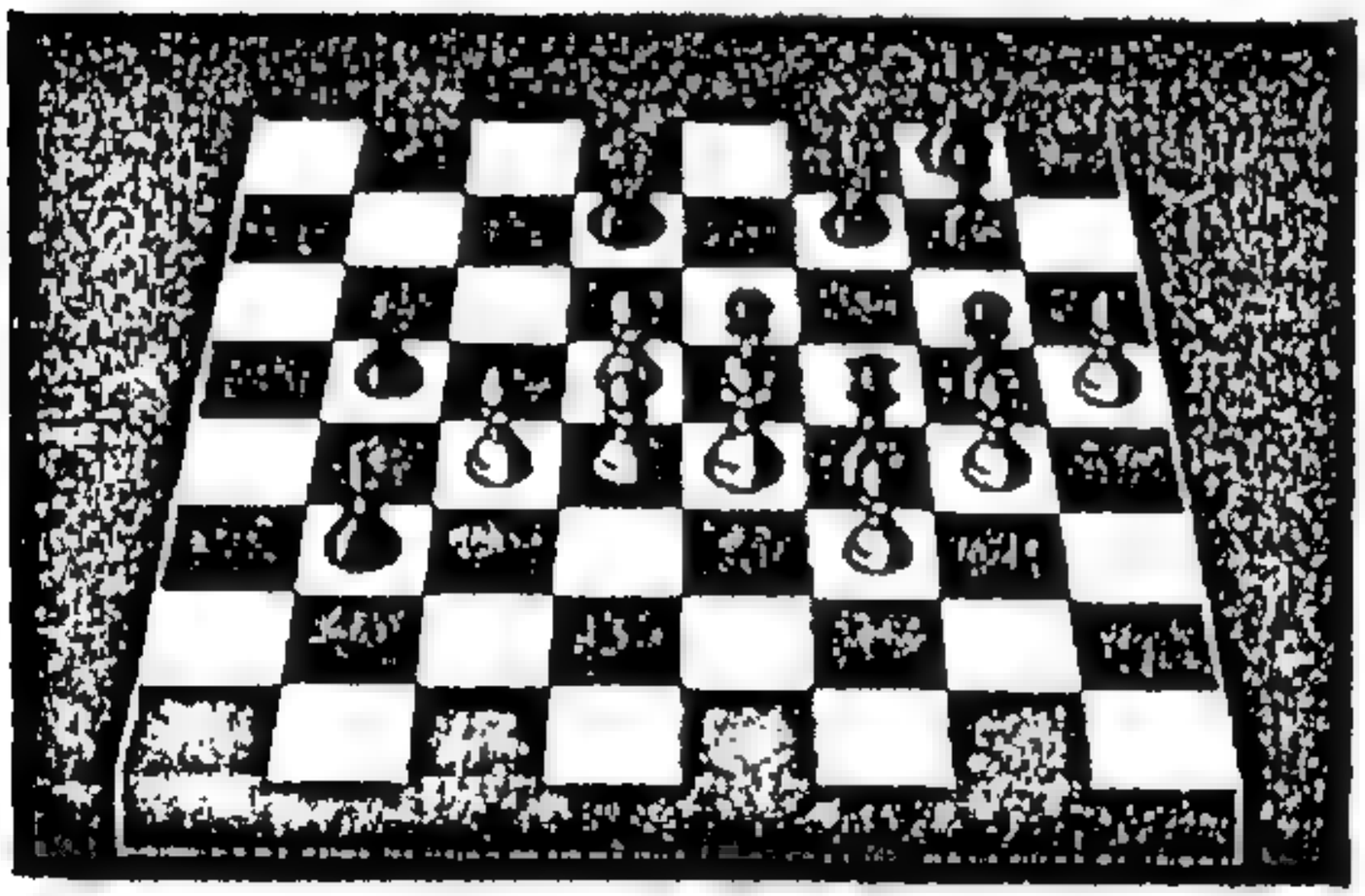


CHECKMATE

9L REGRAS

1	D2-D4		
1	. . .	D7-D5	
2	C2-C4		
2	. . .	C7-C6	
3	C1-F4		
3	. . .	G8-F6	
4	E2-E3		
4	. . .	[illegible]	
5	G1-F3		
5	. . .	E7-E6	
6	B1-C3		
6	. . .	G4xF3	
7	D1xF3		
7	. . .	[illegible]	
8	F1-D3		
8	. . .	D8-A5	
9	E1-B2		
9	. . .	E8-G8	
10	C4-C5		
10	. . .	B4xC3	
11	B2xC3		
11	. . .	B7-B5	
12	C5xB6		
12	. . .	A7xB6	
13	F4-D6		
13	. . .	[illegible]	
14	H1-B1		
14	. . .	B8-B7	
15	D6-B4		
15	. . .	A5-A4	
16	B4-D6		
16	. . .	[illegible]	
17	B1-B4		
17	. . .	A4-A3	
18	F3-F4		
18	. . .	A3-A7	
19	A2-A3		
19	. . .	H7-H6	
20	H2-H4		
20	. . .	D7-F8	
21	B6-C5		
21	. . .	A7-A5	
22	F4-D6		
22	. . .	E8-C8	
23	B6-G3		
23	. . .	F8-D7	
24	C5-E7		
24	. . .	C8-C7	
25	F2-F3		
25	. . .	F6-E8	
26	G3-E1		
26	. . .	D7-E5	
27	E7-C5		
27	. . .	E5xD3	
28	B2xD3		
28	. . .	C7-B7	
29	H4-H5		
29	. . .	E8-F6	
30	G2-G4		
30	. . .	A5-D8	
31	A3-A4		
31	. . .	D8-A5	
32	E1-D1		
32	. . .	A5-A6	
33	A4-A5		
33	. . .	A6-B7	
34	D3-E2		
34	. . .	D7-C7	
35	A5-A6		
35	. . .	C7-H2	
36	E2-D3		
36	. . .	H2-G3	
37	A6-A7		
37	. . .	B7-B7	
38	A1-A6		
38	. . .	B7-C7	
39	A6-A5		
39	. . .	F6-B7	
40	C5-E7		
40	. . .	C7xA7	

41	A5xA7			
41		.	.	ABxA7
42	B3-E2			
42		.	.	A7-A2
43	E2-B3			
43		.	.	E6-E5
44	E3-E4			
44		.	.	G3-F4
45	E7-G5			
45		.	.	H6xG5
46	B4xB5			
46		.	.	C6xB5
47	C3-C4			
47		.	.	F4-H2
48	E4xB5			
48		.	.	A2-A3
49	B1-B3			
49		.	.	A3xB3
50	B3-E4			
50		.	.	H2-F4



CHECKMATE

MICRODRIVES: SU ESTRUCTURA

Los microdrives son la alternativa ofrecida por Sinclair al sistema de almacenamiento en cassette sobre el que ofrece innumerables ventajas tales como control del dispositivo desde el ordenador, mayor velocidad, mayor fiabilidad... Pero en honor a la verdad hay que decir que estos dispositivos son objeto de innumerables criticas al compararlos con las unidades de disco, ya que ofrecen menor capacidad, mayor tiempo de acceso y menos fiabilidad, aunque lógicamente tienen un precio muy interesante, ya que si la configuración del QL ofreciera 2 lectoras de discos el precio subiría considerablemente.

Ciertamente nos hubiera parecido más acertado el diseñar un Spectrum con una o dos unidades de microdrives y no el adosárselas a un QL. No obstante el QL puede trabajar perfectamente con unidades de disco o con disco duro.

En este artículo no vamos a cuestionarnos los pros y los contras de los microdrives, únicamente vamos a exponer una descripción detallada de los mismo.

Los cartuchos de microdrive albergan un bucle de cinta magnetizable de alta calidad. Dicho bucle está dividido en sectores. Una cinta puede dividirse en unos 215 sectores (aprox.); cada sector puede albergar 512 octetos de datos (1/2 Kb.) por lo que la capacidad total de una cinta ronda los 100 Kbytes.

El tiempo que tarda en dar la vuelta completa todo el bucle de cinta de un cartucho cuando lo remueve el motor del QL es de unos 7 segundos. Esto es, cualquier sector de la cinta pasará frente a la cabeza de lectura/escritura una vez cada 7 segundos.

Teóricamente toda la superficie de la cinta es capaz de soportar información, pero en la práctica no es empleada en su totalidad para ser grabada, ya que se dejan unas zonas de "seguridad" entre sectores, para evitar que el grabar o borrar un sector, se pueda interferir en uno de los contiguos. Esto podría ocurrir si un sector comenzara justo a continuación del anterior, ya que si, por ejemplo, se borrara ese sector en otro equipo cuyo motor tuviera una pequeña diferencia de velocidad con el que lo grabó, podría borrarse parte del sector contiguo.

Conviene tener en cuenta que el microdrive es un medio de almacenamiento mas bien frágil, y que por lo tanto es aconsejable tratarlo con cuidado para optimizar su fiabilidad. En principio resultan aplicables los mismos consejos que a cualquier sistema de almacenamiento sobre superficies magnéticas:

- No tocar la superficie magnética con ningún objeto, y si fuera estrictamente necesario por estar la cinta atascada o haberse formado un bucle en la misma procurar hacerlo con un útil liso y limpio (ej. un abrecartas).
- No aproximar los cartuchos de cinta a campos magnéticos, ni guardarlos cerca de ca de los mismos (monitor, impresora, transformador, televisiones, motores...)
- No dejarlos fuera de las fundas protectoras, evitando así que se depositen partículas de polvo sobre la superficie magnética.
- No exponerlos a temperaturas elevadas (dejarlos por ej. en un coche al sol).

Es destacable también la atención que las rutinas de escritura / lectura / verificación del QL prestan a los microdrives en un intento de salvar cualquier error transitorio que pudiera producirse. Así por ejemplo, cada bloque que graba lo verifica cuando vuelve a pasar la cabeza por el mismo (aprox. a los 7 segundos). Si la verificación no se resulta satisfactoria el QL volverá a escribirlos en la misma posición.

Cada bloque de datos va acompañado de un "checksum" de 16 bits, con lo que la posibilidad de que pase inadvertido un error de lectura es de 1/65535.

Si el QL detecta algún error en una operación de lectura o escritura, no abandona la misma sin más; antes de dejarla y mostrar el correspondiente mensaje de error, hace varios intentos, para cerciorarse de que no se trata de un error permanente en la cinta. (4 intentos para una operación de escritura y 8 para una de lectura).

La rutina de formateo, entre otras, tiene la misión de detectar posibles zonas de la banda magnética dañadas, o susceptibles de perder información y anularlas, marcándolas convenientemente para que el QL sepa que no debe utilizarlas. Por eso, una cinta que haya perdido información, puede volver a quedar útil después de formatearla.

Hay que tener en cuenta que con el tiempo la banda magnética se va degradando y comienza a aumentar el número de errores. Cuando esto ocurre deben salvarse los datos en un cartucho nuevo y desecharse el viejo.

TRUCOS

Este programa permite obtener toda la información àtil existente en las cabeceras de los ficheros almacenados en microdrive, se diferencia del comando DIR y del VSTAT del toolkit en que ninguno de ellos extrae toda esta información.

Asi el programa deberà colocarse como procedimiento residente y para llamarlo bastarà con teclear "dcat" una vez cargado. La salida se realiza por el canal 1 y debemos de teclear como parànetro del comando la unidad de la que deseemos obtener el directorio, el parànetro deberà ir entre comillas, asi:

dcat 'mdv1_'

El programa tiene dos bloque claramente diferenciados en primer lugar se realiza la conexión como procedimiento en el primer párrafo del programa principal y posteriormente en PROC_DEF.

Con cada llamada comienza la ejecución en CAT, donde se realiza un anàlisis de los parànetros del comando de forma que puesto que de entrada siempre A3 se encuentra apuntando al comienzo de los parànetros y A5 al final y que la cantidad de octetos asignados por cada parànetro son 8. Asi se comprueba si los hay y en caso de que los haya si son de tipo cadena, en caso contrario se retorna un mensaje de error.

En CAN se realiza la bàsqueda del identificador del canal 1 que està en la tabla de canales del BASIC. En MEB se realiza la apertura de un canal para el directorio del MDV1_ en caso de que no se haya indicado ningún medio especificamente. MMEB realiza la asignación del canal al directorio en el caso de que se haya especificado una unidad. La información se extrae mediante el vector CA_GTSTR y recordando que A1 al comenzar los procedimientos apunta siempre a la pila aritmética de forma que no es preciso tocarlo antes. En este caso la apertura del canal se ha realizado con un TRAP #4 para hacerlo relativo al A6 como se exige en este caso.

En CATEX comienza la ejecución del programa en si, que es un bucle con PRINCIPAL asi se posiciona el puntero en el directorio, se abre la cabecera del fichero que nos indique éste para extraer la información (llamo a OBTENER) se cierra el canal asociado a este fichero se posiciona el puntero en la siguiente zona del directorio y se repite el proceso hasta el fin del fichero.

OBTENER es la subrutina que manda a pantalla toda la información, nombre, tipo (-ejecutable en código màquina, 0- en caso contrario), longitud del programa, del espacio de datos y la fecha.

En el caso de la fecha puesto que antes de la llamada al vector que transforma la palabra larga que la representa en algo legible, es preciso poner A6 a 0 y esto no es posible hacerlo normalmente ya que el A6 es el puntero del BASIC y no debe ser alterado se ha optado por entrar en supervisor (TRAP #0), alterarlo y volver a restaurarlo al finalizar la llamada.

Decir que en este número y siguientes no aparecerà el listado en BASIC ya que se considera que es màs còmoda la solicitud del mismo a la librería.

Isidro Asla.

```
CA_GTSTR  EQU     $115
BP_INIT   EQU     $110
BV_CHBAS  EQU     $30
BV_CHP    EQU     $34
ERR_NO    EQU     -6
IO_OPEN   EQU     1
IO_CLOSE  EQU     2
IO_SSTRG  EQU     7
FS_HEADR  EQU     $47
FS_MDINF  EQU     $45
IO_FSTRG  EQU     3
FS_POSAB  EQU     $42
ASCII     EQU     $30
UT_ERR    EQU     $CC
FS_POSRE  EQU     $43
IO_FBYTE  EQU     1
IO_SBYTE  EQU     5
CN_DATE   EQU     $EC
SD_CLEAR  EQU     $20
SD_SETIN  EQU     $29
SD_SETPA  EQU     $27

* Macros
QDOS      MACRO
          MOVEQ     #\1,D0
          TRAP      #2
          ENDM
INK       MACRO
          MOVE.L    8(A2),A0
          MOVEQ     #-1,D3
          MOVEQ     #\1,D1
          QDOS      SD_SETIN,3
          ENDM
FIN       MACRO
          MOVE.L    (A2),A0
          QDOS      IO_CLOSE,2
          MOVEQ     #0,D0
          RTS
          ENDM
AMANDAR   MACRO
          INK       7
          LEA.L     11,A1
          MOVE      (A1),D2
          MOVEQ     #-1,D3
          LEA.L     2(A1),A1
          MOVE.L    8(A2),A0
          QDOS      IO_SSTRG,3
          INK       0
          ENDM
ENTER     MACRO
          MOVEQ     #$A,D1
          MOVEQ     #-1,D3
          QDOS      IO_SBYTE,3
          ENDM
LIMPIAR   MACRO
          MOVE.L    8(A2),A0
          MOVEQ     #-1,D3
          QDOS      SD_CLEAR,3
          ENDM

Programa principal

          MOVE.W    BP_INIT,A2
          LEA.L     PROC_DEF,A1
          JSR       (A2)
          MOVEQ     #0,D0
          RTS

PROC_DEF  DC.W      1
          DC.W      CAT-*
          DC.B      4,'BCAT'
          DC.W      0
          DC.W      0
          DC.W      0

CAT       MOVEQ     #1,D6
          LEA.L     CANAL,A2
          CMPA.L    A3,A5
          BNE       NMED

          BSR       CAM
          CMP.L     #0,D0
          BNE       FERR
          BSR       MED
          BRA       CATEX

FERR      MOVEQ     #0,D0
          RTS
```

```
KYED
        BSR     CAN
        CMP.L   #0,D0
        BYE     FERR
        MOVEQ   #15,D0
        AND.B   11A6,A3.L1),D0
        SUBQ    #1,D0
        BXE     NOCAB
        MOVE.L  A5,-(A7)
        LEA.L   8(A3),A5
        MOVE.W  CA_GTSTR,A4
        JSR     (A4)
        MOVE.L  (A7)+,A5
        BNE     PRINT_ERR
        MOVEQ   #3,D1
        ADD.W   0(A6,A1.L),D1
        BCLR    #0,D1
        ADD.L   D1,$58(A6)
        MOVEQ   #4,D3
        MOVEQ   #-1,D1
        MOVE.L  A1,A0
        TRAP    #4
        DOS     IO_OPEN,2
        CMP.L   #0,D0
        BNE     PRINT_ERR
        MOVE.L  A0,(A2)
        BRA     CATEX

CAN     MULU    #328,D6
        MOVE.L  BV_CHBAS(A6),A0
        ADDA.L  D6,A0
        CMPA.L  BV_CHP(A6),A0
        BGE     ERR_NUM
        MOVE.L  0(A6,A0.L),D0
        BLT     ERR_NUM
        MOVE.L  D0,A0
        MOVE.L  A0,8(A2)
        MOVEQ   #0,D0
        RTS

MED     LEA.L   DEFMEB,A0
        MOVEQ   #-1,D1
        MOVEQ   #4,D3
        DOS     IO_OPEN,2
        CMP.L   #0,D0
        BNE     APRINT_ERR
        MOVE.L  A0,(A2)
        MOVEQ   #0,D0
        RTS

NOCAB   MOVEQ   #-15,D0
        BRA     PRINT_ERR

PRINT_ERR
        MOVE.W  $CA,A2
        JSR     (A2)
        MOVEQ   #0,D0
        RTS

ERR_NUM MOVEQ   #ERR_NO,D0

APRINT_ERR
        MOVE.W  $CA,A2
        JSR     (A2)
        RTS
```

```
}

CATEX
        MOVE.L    (A2),A0
        MOVEQ     #$0F,D6
        MOVEQ     #$0F,D1
        MOVEQ     #-1,D3
        BIOS      FS_POSAB,3
        LIMPIAR

PRINCIPAL MOVE    #-1,D3
        MOVEQ     #0,D1
        MOVE.L    (A2),A0
        BDOS      IO_FBYTE,3
        MOVE      D1,D5
        MOVEQ     #-1,D3
        LEA.L     FICHERO,A4
        LEA.L     7(A4),A1
        BIOS      IO_FSTRG,3
        ADD       D5,D5
        MOVE.W    D5,(A4)
        MOVEQ     #-1,D1
        MOVEQ     #0,D3
        LEA.L     FICHERO,A0
        BIOS      IO_OPEN,2
        MOVE.L    A0,4(A2)
        MOVEQ     #64,D2
        MOVEQ     #-1,D3
        LEA.L     BUFFER,A1
        BDOS      FS_HEADR,3
        BSR       CONTENER
        MOVE.L    4(A2),A0
        BIOS      IO_CLOSE,2
        ADD.L     #64,D6
        MOVE.L    D6,D1
        MOVEQ     #-1,D3
        MOVE.L    (A2),A0
        BDOS      FS_POSAB,3
        CMP       #0,D0
        BNE       ACAB
        BRA       PRINCIPAL

ACAB
                  FIN

;Rutina de conversion

DECIMAL   MOVEQ   #0,D1
          MOVEQ   #0,D4
          LEA.L   BMEM,A1
          LEA.L   TEMP,A0
          MOVEQ   #4,D3
          MOVEB   #ASCII,D0
          MOVE    #10000,D1
BUCLE     DIVU    D1,D2
          CMP     #4,D3
          BNE     SALTO
          MOVE.L  D2,(A0)
          AND.L   #$FFFF,D2
          MULU    #$A,D2
          MOVE.W  D2,D4
          SWAP    D2
          AND.L   #$FFFF,D2
          MULU    #10000,D2
          MOVE.W  (A0),D5
          ADD.L   D5,D2
          BRA     CFIN
SALTO     MOVE.W  D2,D4
          SWAP    D2
          AND.L   #$FFFF,D2
          DIVU    #$A,D1
CFIN      ADD.L   D0,D4
          MOVE.B  D4,(A1)+
          DBRA    D3,BUCLE
          ADD.L   D0,D2
          MOVE.B  D2,(A1)+
          RTS
```

```
OBTENER  LEA.L    BUFFER,A4
         AMANDAR  PRO
         MOVE     $0E(A4),D2
         MOVEQ    #-1,D3
         LEA.L    $10(A4),A1
         QDOS     IO_SSTRG,3
         ENTER
         AMANDAR  TIPO
         MOVE.B   5(A4),D1
         ADD      #ASC(1,D1
         MOVEQ    #-1,D3
         QDOS     IO_SBYTE,3
         ENTER
         AMANDAR  PLON
         MOVE.L   (A4),D2
         BSR      DECIMAL
         MOVEQ    #6,D2
         MOVEQ    #-1,D3
         LEA.L    BMEM,A1
         MOVE.L   8(A2),A0
         QDOS     IO_SSTRG,3
         ENTER
         AMANDAR  DLON
         MOVE.L   6(A4),D2
         BSR      DECIMAL
         MOVEQ    #6,D2
         MOVEQ    #-1,D3
         LEA.L    BMEM,A1
         MOVE.L   8(A2),A0
         QDOS     IO_SSTRG,3
         ENTER
         AMANDAR  FE
         TRAP     #0
         MOVE.L   A6,-(A7)
         SUBA.L   A6,A6
         MOVE.L   $34(A4),D1
         LEA.L    PILA_BASE,A1
         MOVE.W   CN_DATE,A4
         JSR      (A4)
         MOVE.L   (A7)+,A6
         ANDI.W   #$DF,SR
         MOVEQ    #-1,D3
         MOVE     (A1)+,D2
         MOVE.L   8(A2),A0
         QDOS     IO_SSTRG,3
         ENTER
         RTS

BUFFER   DS.B     64
BUFFER_TOP
BEFMEN   DC.W     5
         DC.B     'MBV1_',0
PILA_TOP DS.B     32
PILA_BASE
FICHERO  DS.W     1
         DC.B     'MBV1_'
         DS.B     39
BMEM     DS.B     8
TEMP     DS.B     16
CANAL    DS.L     3
PRO      DC.W     10
         DC.B     'Programa: '
TIPO     DC.W     9
         DC.B     '   Tipo: ',0
PLON     DC.W     25
         DC.B     '   Longitud de programa: ',0
DLON     DC.W     21
         DC.B     '   Espacio de Datos: ',0
FE       DC.W     10
         DC.B     '   Fecha: '

         END
```

```
Programa: e4
   Tipo: 1
   Longitud de programa: 021548
   Espacio de Datos: 004800
   Fecha: 1961 Jan 01 00:00:00
Programa: asma_ovl
   Tipo: 0
   Longitud de programa: 007180
   Espacio de Datos: 000000
   Fecha: 1961 Jan 01 00:00:00
Programa: asmb_ovl
   Tipo: 0
   Longitud de programa: 007452
   Espacio de Datos: 000000
   Fecha: 1961 Jan 01 00:00:00
```

COMENTARIO DE PROGRAMAS

PROGRAMA: QL Assembly Language Microdrive Toolkit, Versión 1.00
EDITOR: COMPWARE
DIRECCION: 57 Repton Drive, Haslington, Crewe, CW1
PRECIO: 29€

Se trata de otra herramienta más que presenta Compware para los programadores de lenguaje ensamblador en el QL.

En un microdrive corriente se nos presentan tres títulos:

- YOUR_ASM
- MDKIT_ASM
- BACKUP

Los dos primeros son programas de código fuente en ensamblador, y el tercero es, como se puede adivinar, un programa en basic para realizar la siempre aconsejable copia de seguridad del microdrive.

El producto va dirigido a programadores con unos ciertos conocimientos en el lenguaje ensamblador.

El programa principal es MDKIT_ASM, que consiste en una serie de rutinas en ensamblador dirigidas al manejo de microdrives, a la grabación, verificación y lectura en el ámbito de sectores. Cada rutina lleva una etiqueta característica con la que se invoca. Las rutinas están ampliamente comentadas (con mas de mil líneas a este propósito en el mismo programa), con lo que se convierte a la vez en un elemento didáctico y en una importante herramienta de trabajo con microdrives.

A las rutinas se les pasan los datos normalmente en los registros bajos, recibiendo respuestas en estos mismos (entre estas respuestas los errores posibles).

Las principales rutinas son:

- m_format

Formatea un microdrive con un número pasado en el registro de datos D1. El puntero al nombre del medio se pasa en A0. El tiempo máximo es de 30s.

- m_bformat

Formatea un microdrive idénticamente a m_format, pero las direcciones se manejan relativas al registro A6.

- m_rrad

Lee la cabecera de sector, pasándole el número de microdrive en el registro D3. En A1 se pasa un puntero a un buffer de al menos 14 bytes que se llenara con la cabecera de sector. Tarda como máximo 3s.

- m_bhread

La misma rutina anterior con las direcciones relativas al registro de direcciones A6.

- m_sread

Lee un sector de un microdrive, se le pasa en D3 el número de sector, en D4 el numero de microdrive, y en A1 un puntero al buffer de al menos 512 bytes. Tarda unos 3.5

segundos, y unos 128 si le fallan los ocho intentos que hace (se puede cambiar).

- m_sverify

Verifica un sector de un microdrive comparàndolo con una zona de memoria que se pasa en A1. Se le pasa en D3 el nùmero de sector, y en D4 el numero de microdrive. Suele tardar 3.5 segundos, 16 segundos màximo. Solo realiza un intento.

- m_swrite

Escribe un sector en el microdrive. El nùmero de fichero se pasa en D1, el numero de bloque del fichero en D2, el numero de sector en D3, el numero de microdrive en D4 y el puntero al buffer de donde se escribirà 512 bytes en A1. Suele costarle unos 10.5s, 144s màximo.

Esta rutina invoca a la de verificaciòn.

- m_swriterov

Esta rutina hace lo mismo que la anterior, con la salvedad de que no verifica los datos, con el ahorro de tiempo que ello conlleva, pero tambien con el peligro de perder datos. Suele tardar unos 3.5s en realizar la tarea, màximo 128s.

- Ademàs de estas rutinas, aparecen otras que son invocadas desde èstas, como son las de seleccionar un microdrive (md_mdsel), deseleccionar un microdrive (md_mddsel), leer el siguiente sector (m_qsread), verificar el siguiente sector (m_qsverify), escribir el siguiente sector (m_qswrite), encontrar un sector (m_sfind)...

El programa, al estar en ensamblador fuente, es completamente modificable y expandible con propias rutinas (¿por que no?!.

Al principio de la utilidad se definen tres constantes especialmente interesantes, la primera es el nùmero de veces que se intentarà la lectura y escritura de un sector (en caso de fallos), que esta puesto a ocho (se puede separar el numero de intentos en lectura y escritura cambiando las instrucciones 'move' que tratan esta variable).

La segunda constante hace referencia al nùmero de intentos en la verificacion, esta puesta a un solo intento.

Se proporciona en el microdrive un ultimo programa, el 'YOUR_ASM', en codigo fuente ensamblador, que incluye, por medio de un 'GET 'mdkit_asm'", todas las rutinas anteriores, y les proporciona las constantes necesarias. Este programa esta diseñado (el 'your_asm') para que se inserte en el el programa del usuario, con las llamadas que necesite a las rutinas. Es un esqueleto del programa ensamblador del usuario.

La documentaciòn que acompaña al microdrive es màs que suficiente. El manual consiste en un cuadernillo tamaño cuartilla de color rojo, en el que se incluyen unas fichas con un pàrrafo de explicaciòn de cada una de las rutinas. Ademàs se incluye en este manual un par de hojas con explicaciones sobre el funcionamiento interno de la màquina respecto a los microdrives, y sobre la divisiòn de espacio y estructura de datos en el mismo. No se puede olvidar aqui que el programa lleva mas de mil lineas de comentarios que pueden proporcionar al lector unos mayores conocimientos sobre el uso de los microdrives. El programa es perfectamente inteligible.

Segùn se dice en el manual, el programa funciona en los ensambladores de Metacomco y Alder sin modificaciòn.

Javier Boira

NOTIFICACIONES

LIBRERIA DE PROGRAMAS

Este mes se han incorporado a la librería los siguientes programas:

15.- Efemérides_astronómicas. Longitud: 18931 bytes.

Este programa proporciona tras la introducción de los datos solicitados la hora en Greenwich, la declinación, altura y azimut para los planetas y Aries. Para el Sol además las horas en T.U. de salida y puesta y los crepúsculos civil y náutico. Para las estrellas el Angulo sidéreo, su horario en el lugar, declinación, altura y azimut.

La precisión obtenida en los cálculos es de:
Para el Sol y las estrellas mejor de 1 minuto de arco.
Para Venus y Marte mejor de 2 minutos.
Para Júpiter y Saturno mejor de 3 minutos.
Para la Luna mejor de 5 minutos de arco.

16.- Diseny_cad Longitud: 17852 bytes.
Qprint_prt Longitud: 290 bytes.

Este programa permite el diseño de circuitos eléctricos. La estructura del programa es semejante a la de los programas de Psion, con menú en la parte superior de la pantalla y fichero de ayuda pulsando F1.

17.- Mapa_ch Long. 467 bytes.
pr_ch Long. 5088 bytes.
xy_ch Long. 11555 bytes.
ca_ch Long. 187 bytes.
no_ch Long. 4657 bytes.
gráfico Long. 32768 bytes.

Este programa nos proporciona un mapa autonómico del Estado español y permite obtener dibujos de cada una de las autonomías así como la situación de gran cantidad de ciudades y la distancia entre ellas.

18.- cat_asm Long. 7190 bytes.
cat_com Long. 1124 bytes.

Este programa corresponde al listado que se incluye en este mismo número en ensamblador.

QLave.

Muchas casas de software protegen sus programas contra los 'piratas' aprovechando el número aleatorio que llevan los cartuchos de microdrive formateado en todos los sectores. Dichos programas, antes de ejecutarse comprueban que el cartucho en el que están grabados tenga un determinado número grabado. De no ser así, sabe que se trata de una copia pirata, y por tanto no llega a ejecutarse.

Este sistema de protección aunque está encaminado a reducir la piratería, plantea problemas a los usuarios, ya que el deterioro o extravío del cartucho original puede tener consecuencias catastróficas.

El programa que presentamos en esta sección ofrece 3 opciones:

- Lectura del número aleatorio grabado en un cartucho.
- Formateo de un cartucho con un determinado número.
- Copia de un cartucho con el mismo número que el original.

Somos conscientes que al igual que este programa puede ser muy útil para los socios del Club, también es una herramienta de trabajo fantástica para los piratas; pero resulta evidentemente nosotros abordamos el problema desde el punto de vista de un Club de Usuarios, ofreciendo la posibilidad de hacer <u>copias de seguridad</u> que no sólo son perfectamente legales, sino aconsejadas por todas las casas de software.

Esperemos que lleguen a Vd. con algún sistema de protección que no presente molestias ni problemas al usuario.

El programa cuyo listado ofrecemos a continuación está completamente desarrollado en SuperBasic, y podemos decir que las líneas que llevan a cabo todo el trabajo son: 540, 760, 770, 910,920.

En estas líneas se leen las direcciones de memoria a las que señalan los punteros de las direcciones 164076 y 164188, con un desplazamiento de 32. En dichas direcciones almacena el QL el último número aleatorio leído en el microdrive 1 y 2 respectivamente.

Asimismo, cuando el QL formatea un cartucho extrae de la dirección 163895 el número que va a grabar en todos los sectores del microdrive.

Por eso el programa para leer el número de un cartucho lo que hace es pedirle primero el directorio y una vez que lo ha hecho, y por lo tanto ha almacenado el aleatorio en una de las direcciones mencionadas, lo lee de la memoria. Para que no aparezca el directorio en la pantalla al ir a leer el número de un cartucho, este se envía a una ventana abierta en la línea 110, cuyas dimensiones no permiten verlo.

Inversamente cuando se trata de formatear un cartucho con un determinado número, lo que se hace es 'pokear' primero en la dirección 163885 el número deseado y acto seguido pedirle al QL que lo formatee. Cuando la rutina de formateo del QL accede a la memoria para leer el 'aleatorio' con que va a formatear el cartucho lo que hace es tomar el número que hemos introducido nosotros.

Conviene tener en cuenta que el número aleatorio tiene un tamaño de 2 bytes, por lo que tiene que ser un número sin signo comprendido en el rango 0 / 65535 o un número con signo del rango -32768 / +32767.

```
100 REMark backup - Juan Palacio
110 OPEN #12, scr_5x5a1x1 : GO TO 170
120 AT 1,75 : PRINT "pulsa una tecla para continuar"
130 tecla 0,150
140 REMark ####################
150 REMark ## PRESENTACION ##
160 REMark ####################
170 WINDOW 512,256,0,0
180 PAPER 0 : INK 7 : CLS
190 OPEN #11, scr_85x23a200x0
200 BORDER #11, 1,4
210 CSIZE #11,3,1
220 PRINT #11, "Slave""
230 AT 3,18: PRINT "Club Espaol Independiente de Usuarios del QL"
240 AT 4, 20 : PRINT "Apartado de correos 493. 50080 - Zaragoza"
250 LINE 0,80 TO 150,80
260 WINDOW 512,220,0,56
270 AT 6,15 : PRINT "1.- Leer el nmero aleatorio de un cartucho."
280 AT 8,15 : PRINT "2.- Formatear un cartucho con un determinado nmero."
290 AT 10,15 : PRINT "3.- Copia de seguridad con el mismo nmero que el cartucho fuente."
300 PAPER 7 : INK 0 : AT 17,33
310 PRINT " Pulsa opcion "
320 INK 7 :PAPER 0
330 tecla 49,51
340 res=CODE(res$)-48
350 CLS
360 ON res GO SUB 720,830,410
370 GO TO 120
380 REMark ##########################
390 REMark ## COPIA DE SEGURIDAD ##
400 REMark ##########################
410 AT 8,7 : PRINT "Introduce el cartucho fuente en mdv1 y el cartucho destino en mdv2"
420 AT 17, 25 : PRINT "Pulsa una tecla para continuar"
430 tecla 0,250
440 CLS
450 CSIZE 1,1
460 AT 3,10 : PRINT "SE FORMATEARA EL CARTUCHO DEL MICRODRIVE 2"
470 CSIZE 0,0
480 AT 17,30 : PRINT "Pulsa '0' para continuar"
490 tecla 48,48
500 CLS
```

```
510 DIR #12, mdv1_
520 AT 1,25 : PRINT"Numero clave del cartucho: ";PEEK_W(PEEK_L(144076)+32)
530 PRINT "Formateando MDV2_..."
540 POKE_W 163236,(PEEK_W(PEEK_L(164096)+32))-1:FORMAT mdv2_
550 OPEN_NEW #13,mdv2_dir
560 DIR #13,mdv1_
570 CLOSE #13
580 OPEN_IN #13,mdv2_dir
590 INPUT #13,a$;a$
600 REPeat bucle
610 IF EOF(#13):EXIT bucle
620 INPUT #13,a$
630 PRINT "COPIANDO MDV2_";a$
640 COPY "MDV1_"&a$ TO "MDV2_"&a$
650 END REPeat bucle
660 CLOSE #13 : CLOSE #11
670 DELETE mdv2_dir
680 RETurn
690 REMark ##################
700 REMark ## LECTURA NUMERO ##
710 REMark ##################
720 AT 8,16: PRINT "Pon el cartucho que quieras leer en      1"
730 AT 7,45: PRINT "mdv1_ y pulsa '1'."
740 AT 9,46: PRINT "mdv2_ y pulsa '2'."
750 tecla 47,50
760 IF res$="1" THEN DIR #12,mdv1_ : codigo=PEEK_W(PEEK_L(164096)+32)
770 IF res$="2" THEN DIR #12,mdv2_ : codigo=PEEK_W(PEEK_L(164100)+32)
780 AT 17,28 : PRINT "El numero clave del cartucho es: ";codigo
790 RETurn
800 REMark ##########################
810 REMark ## FORMATEAR CON NUMERO ##
820 REMark ##########################
830 AT 2,5 : INPUT "Numero con que deseas formatear el cartucho: ",num
840 CLS
850 AT 2,5 : INPUT "Nombre del cartucho: ";nom$
860 CLS
870 AT 8,16 : PRINT "Introduce el cartucho en      1"
880 AT 7,35 : PRINT "mdv1_ y pulsa '1'."
890 AT 9,35 : PRINT "mdv2_ y pulsa '2'."
900 tecla 49,50
910 IF res$="1 THEN POKE_W 163836,num : FORMAT "mdv1_"&nom$
920 IF res$="2 THEN POKE_W 163836,num : FORMAT "mdv2_"&nom$
930 CLS
940 RETurn
950 DEFine PROCedure tecla (x,y)
960 res$=INKEY$(-1) : IF res$="" THEN got 1030
970 IF CODE(res$)<x OR CODE(res$)>y THEN GO TO 960
980 END DEFine tecla
```

MICRODRIVES: GESTION DEL SO

El almacenamiento de la información en los microdrives se se realiza en los ficheros; éstos poseen al comienzo de los mismos una cabecera de 64 octetos que no es accesible desde el BASIC normalmente. La información es la siguiente:

- $00 Longitud del fichero.
- $04 Clave de acceso al fichero.
- $05 Tipo de fichero.
- $06 En el caso de que se trate de programas de tipo 1 los 4 primeros octetos dan la longitud del espacio de datos del programa.
- $0E Longitud del nombre del fichero.
- $10 Nombre del fichero.
- $34 Fecha actualizada.
- $38 Fecha de referencia.
- $3C Fecha de copia.

En lo que se refiere a la Fecha actualizada está implementado en las versiones MG y posteriores. La clave de acceso tampoco es rellenada por el sistema.

Es toda esta información la que se extrae mediante el programa en ensamblador publicado en este mismo número.

En lo que se refiere al directorio decir que es el fichero 0 del microdrive y contiene copias de las cabeceras de sector indicadas anteriormente, aunque las copias en el directorio no son completas así unicamente se guarda la longitud del fichero así como el nombre de éste. Este es el motivo por el que un programa de utilidad como el publicado en este número precise de la lectura de las cabeceras de cada uno de los ficheros componentes del directorio. El proceso de borrado de un fichero vía comando DELETE borra del directorio la longitud del fichero y la longitud del nombre del mismo, de igual modo son colocados a $F0 (es decir como libres) los sectores que lo componen en el mapa de sectores. De forma que un programa como el doctor cartucho debe recomponer los datos del directorio y el mapa de sectores. Una imagen de una parte de un directorio se puede ver en la figura 1.

Los ficheros están situados en los distintos sectores en que se formateó el microdrive y contienen los bloques que contienen la información de los ficheros. Cada sector tiene una cabecera de 14 octetos de longitud.

- $00 Indicador de cabecera de sector $FF.
- $01 Número de sector.
- $02 Nombre del medio.
- $0B Número aleatorio de 16 bit.(se trata del número asignado en formateo y que sirve de protección a los programas comerciales)

Posteriormente cada bloque contenido en el sector posee una cabecera de dos octetos.

- $00 Número de fichero.

- $01 Número de bloque.

El sector 0 contiene el mapa de sectores del medio como puede verse en la figura 2. Así aparecen los dos octetos de cada bloque por sector donde $F8 representa el mapa del microdrive, $FD se refiere a un sector libre y $FF indica un sector dañado. Los demás de $00 a $FE se refieren a números de fichero. El segundo octeto es el número de bloque que ocupa este sector dentro del fichero.

En el QL la gestión de los microdrives la llevan a cabo los controladores de dispositivo de directorio, nosotros podemos conectar otros controladores de dispositivo dependiendo del 'hard' que pretendamos enchufar, el controlador de los microdrives está en la ROM. El proceso que se sigue viene determinado por el mediador de acceso que se encarga de la comunicación con el subsistema de I/O del QL y el mediador físico que se encarga de la conexión entre el controlador del dispositivo y el hard del medio que queremos controlar.

Una vez que pretendemos poner en marcha un microdrive, bien sea pidiendo su directorio, abriendo un fichero o de cualquier otra forma el QDOS crea lo que se llama un bloque de definición física para el microdrive que se use. Estos dos bloques aparecen apuntados por la variable del sistema operativo, SV_FSDEF. La primera palabra larga que nos encontramos en la dirección almacenada por esta variable se refiere al bloque del microdrive 1 y la segunda al del microdrive 2. Estas direcciones son con frecuencia las mismas pero ocasionalmente pueden variar, por lo que es buena política mirar antes la dirección que nos indique SV_FSDEF.

Si tomamos pues como origen la dirección dada por la primera palabra antes referida podemos encontrar los siguientes datos almacenados:

- $10 FS_DRIVR que apunta a la conexión del mediador de acceso para el controlador.
- $14 FS_DRIVN número de microdrive.
- $15 octeto reservado.
- $16 FS_MNAME nombre dado al cartucho al formatear.
- $20 número aleatorio de formateo.
- $22 FS_SPARE número de ficheros abiertos.
- $28 comienzo de una copia del mapa de sectores del microdrive en curso.

Un ejemplo de esto puede verse en la figura 3.

Cada vez que se abre un fichero de microdrive se le debe asignar un canal. Así se realiza una entrada en la tabla de canales del sistema operativo. La situación de la tabla de canales viene indicada por las variables del QDOS:

- $28074 SV_CHPNT apuntador al último canal comprobado.
- $28078 SV_CHBAS apuntador a la base de la tabla de canales.
- $2807C SV_CHTOP apuntador a lo alto de la tabla de canales.

Un detalle de una tabla de canales puede verse en la figura 4. En ella podemos ver que cada entrada es de cuatro octetos y nos da la dirección de la memoria donde se encuentra el bloque de definición del canal. Ver que para todos los canales abiertos el primer octeto está a cero, cuando para los no abiertos está a $FF en el caso de que cerremos

alguno de estos canales de la tabla no se borrará la dirección sino que simplemente se colocará este primer octeto a $FF.

Así pues se crea un bloque de definición de canal que posee los siguientes datos (respecto de la dirección dada por la tabla).

- $00 CH_LEN Longitud del bloque de definición.
- $04 CH_DRIVR Dirección del controlador.
- $08 CH_OWNER Job propietario.
- $0C CH_RFLAG dirección que se manda cuando se libera el espacio.
- $10 CH_TAG Identificador del canal.
- $12 CH_STAT Estado 0= OK. / -1= Ref. Al absoluta / $80= Ref. Al relativo a A6 / neg= esperando.
- $13 CH_ACTN Acción almacenada para el job que espera.
- $14 CH_JOBWT Identificador del job esperando por I/O.
- $18 FS_NEXT Conexión al siguiente canal de fichero del sistema.
- $1C FS_ACCES valor de D3 en la llamada de OPEN.
- $1D FS_DRIVE Identificador de la unidad.
- $1E FS_FILNR Número de ficheros en la unidad.
- $20 FS_NBLOK Número de bloque que contiene el siguiente octeto.
- $22 FS_NBYTE Siguiente octeto del bloque.
- $24 FS_EBLOK Número del bloque que contiene el octeto después del EOF.
- $26 FS_EBYTE Octeto después del EOF.
- $28 FS_CBLOK Apuntador a la tabla de bloques depend.
- $2C FS_UPDT Puesto si se ha actualizado el fichero.
- $32 FS_FNAME Nombre del fichero.
- $58 FS_SPARE 72 octetos libres.

Un ejemplo de este tipo de bloques de definición puede verse en la figura 5.

Hemos visto que FS_CBLOK apunta a la tabla de bloques dependientes. Esto es así ya que el sistema usa la memoria libre del QL como memoria intermedia de lectura de los microdrives. La asignación de esta memoria se realiza por lo que se llama bloques dependientes. Cada bloque asignado implica una entrada en la tabla de bloques de 8 octetos:

- $00 BT_STAT Es un octeto de estado:
  0= imposible / 1= bloque vacio / X3= es fichero/ X7= bloque modificado / X9= espera lectura / XB= espera verif. (X representa el num de microdrive)
- $01 BT_PRIOR Para algoritmos dependientes.
- $02 BT_SECTR Sector fisico en el medio.
- $04 BT_FILNR Número de fichero.
- $06 BT_BLOCK Número de bloque de mdv_.

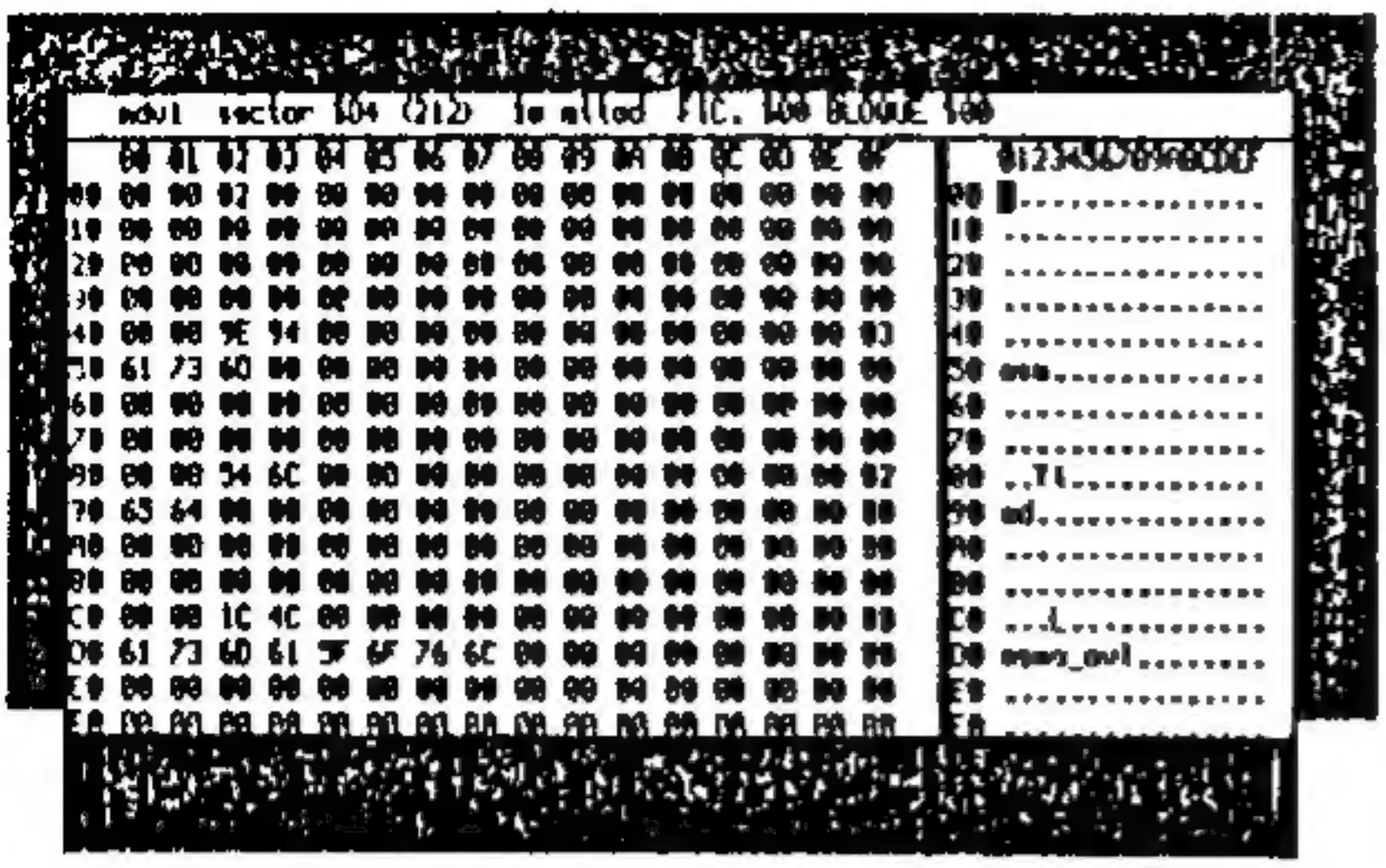


Figura 1. Detalle de un directorio.

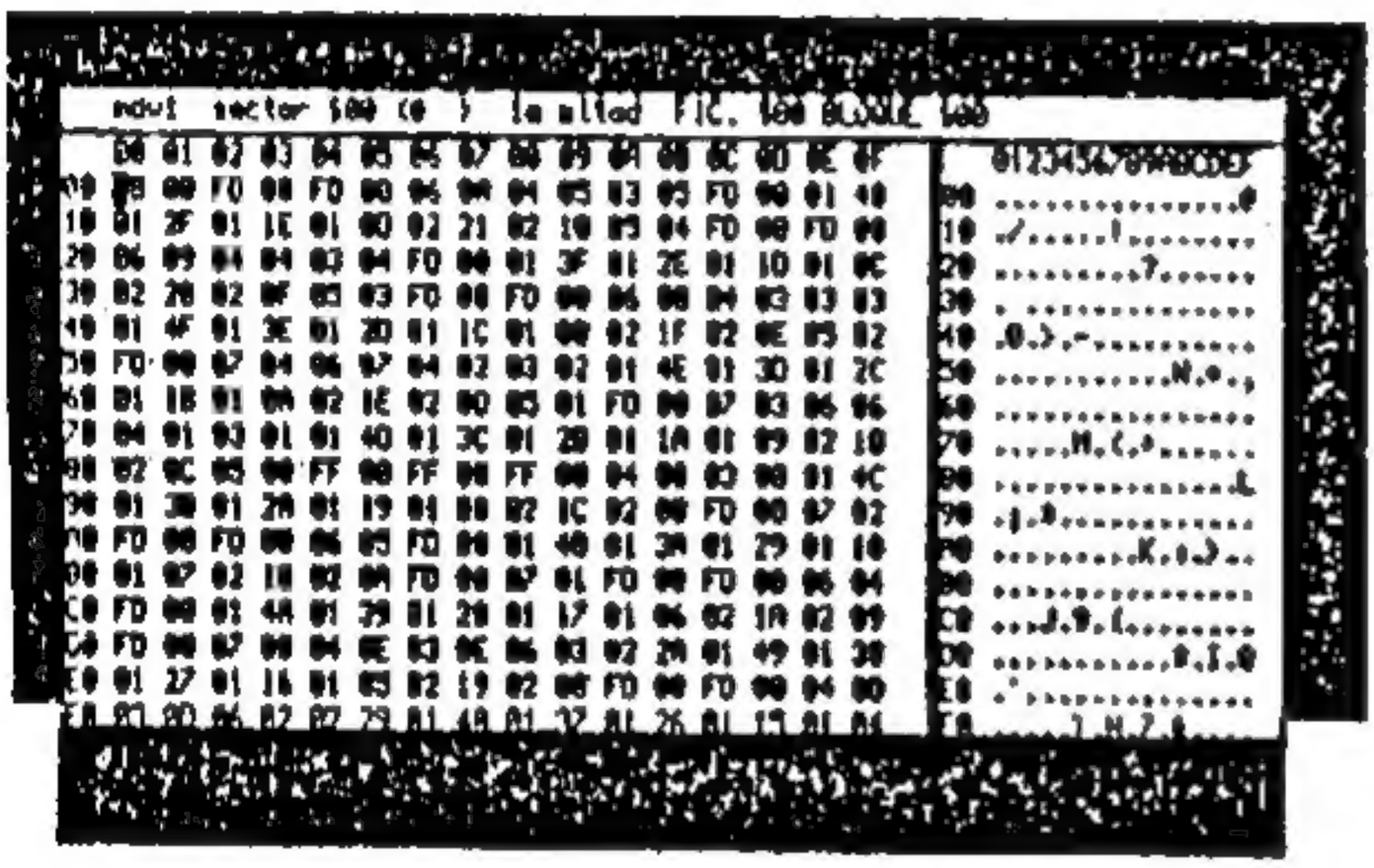


Figura 2. Detalle del mapa de sectores.

```
00029350 00 02 8E 00 00 02 8F 00 ........
00029358 00 02 90 00 00 02 95 80 ........
00029370 00 02 91 00 00 02 96 00 ........
00029378 00 02 97 40 00 02 97 80 ...@....
00029380 00 02 98 20 00 02 99 10 ... ....
00029388 FF 00 00 00 FF 00 00 00 ........
00029390 FF 00 00 00 FF 00 00 00 ........
00029398 FF 00 00 00 FF 00 00 00 ........
```

Figura 3. Bloque de definición de canal de fichero.

```
000291A0 00 00 04 30 00 00 00 00 ...0....
000291A8 00 00 00 00 00 00 00 00 ........
000291B0 00 00 12 34 01 C0 40 4F ...4..@O
000291B8 4E 49 54 4F 52 20 20 20 NITOR
000291C0 4F 92 00 00 00 00 00 00 O.......
000291C8 FB 00 05 13 02 0B 05 02 ........
000291D0 18 09 12 07 03 09 10 0B ........
000291D8 0F 10 0E 02 0B 01 FF 00 ........
000291E0 FF 00 FF 00 05 12 02 0A ........
000291E8 05 01 17 00 06 00 03 08 ........
000291F0 14 00 10 0A 0F 0F 0E 01 ........
000291F8 0B 00 18 08 12 06 05 11 ........
00029200 02 09 05 00 FD 00 19 01 ........
00029208 03 07 0E 11 10 09 0F 0E ........
```

Figura 3. Bloque de definición física.

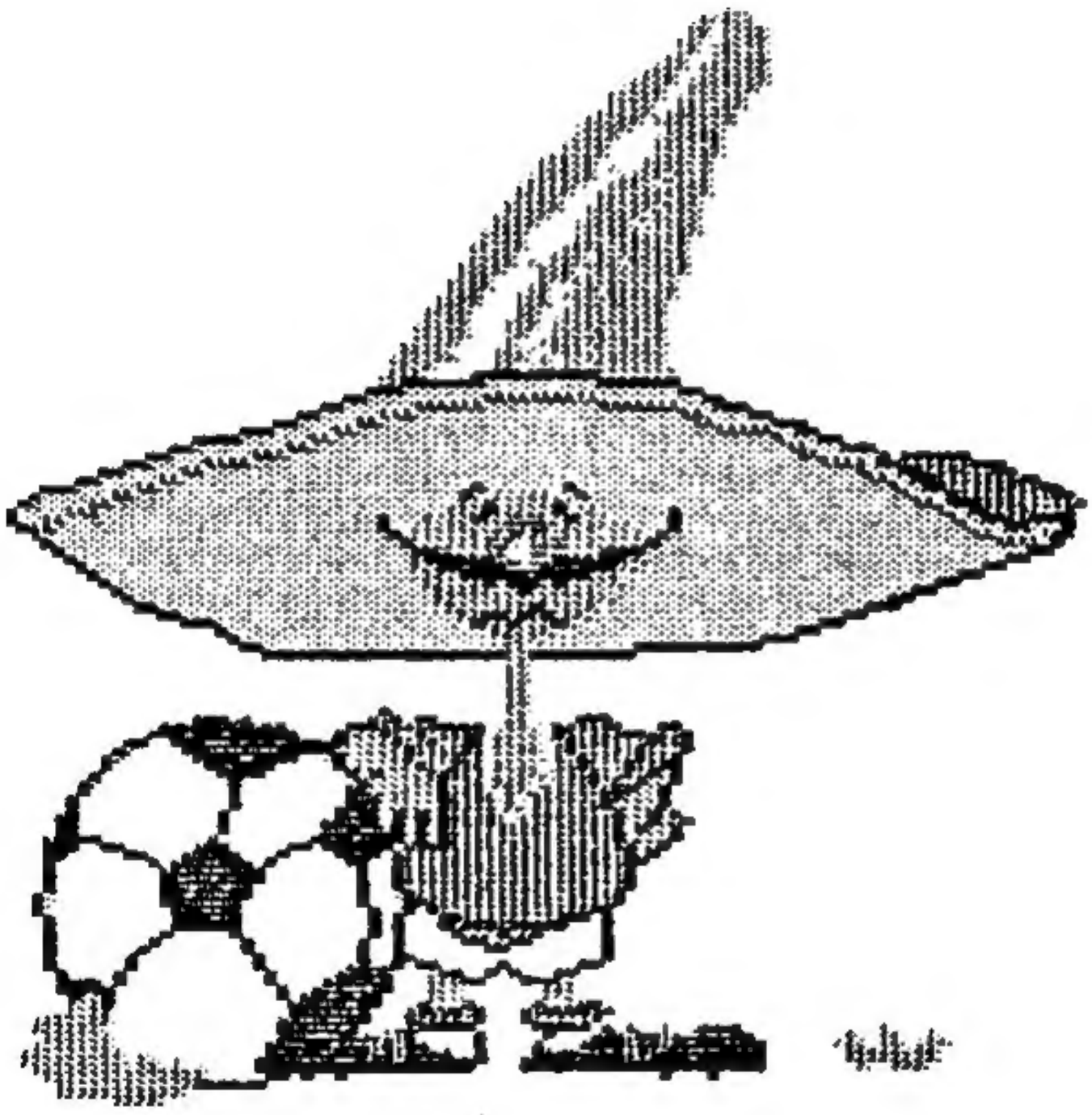
```
00029910 00 00 00 A0 00 00 12 34 .......4
00029918 00 00 00 00 00 02 8B 84 ........
00029920 00 08 00 00 00 00 00 00 ........
00029928 00 00 00 00 00 00 00 05 ........
00029930 00 00 00 40 00 20 01 28 ...@. .(
00029938 00 02 87 88 00 00 00 00 ........
00029940 00 00 00 05 63 31 6D 6F ....c1mo
00029948 6E 00 00 00 00 00 00 00 n.......
00029950 00 00 00 00 00 00 00 00 ........
00029958 00 00 00 00 00 00 00 00 ........
00029960 00 00 00 00 00 00 00 00 ........
```

Figura 4. Tabla de canales del sistema.

CORREO DE LOS SOCIOS

Me gustaría ponerme en contacto con alguien que posea un compilador de LOGO, así como con alguien que posea una rutina en C/M que permita realizar COPYS de pantalla en la SEIKOSHA GP-700

E.J. Sanchis Borrel
C/ Cid n-45 Entresuelo
ALCOY Alicante.



Desearía ponerme en contacto con usuarios del QL, preferiblemente si son residentes en Buenos Aires.
Mis señas son:

Rubén Zeida
Avda. José M. Moreno 453
1424 Buenos Aires.
ARGENTINA (Socio n. 187)

SUMARIO

1.- Portada

2.- Informacion sobre el club

3.- Editorial

5.- Novedades

7.- Preguntas

10.- ...Sobre los Joysticks

14.- Duelo ÀL..
...Amstrad

18.- Microdrives:
- Estructura

20.- Trucos y rutinas
- Cat

25.- Comentario de programas
- ÀL ast. toolkit

27.- Notificaciones

28.- Copias de Seguridad

31.- Microdrives:
- Gestion del SO

35.- Correo de los suscritos

36.- Indice